PAPER — PROGRAMMING LANGUAGES IN PRACTICE

COMP 3010 — ORGANIZATION OF PROGRAMMING LANGUAGES

1. Paper Topic

The paper for Organization of Programming Languages asks you to investigate independently into a topic that interests you about the historical, practical, and human aspects of programming languages used and developed by people over the years. For your subject, choose one of the following two options to write about (for approximately 8 double-spaced pages):

- ***Depth***

  Choose one specific facet of programming language design to investigate, such as type systems, security, memory management, modularity, safety, expressiveness, alternative semantics (like constraint solving) *etc.*
  (1) Introduce and define your chosen facet (at least one page).
  (2) Discuss its development in the field of programming languages by punctuating its history with particular examples of innovations and deployments, and the context which motivated people to develop them. (Each such section should probably be at least a page, maybe more.)
  (3) End with a brief summarization (roughly a page) of open issues and ideas that you have for developing it (this last part is the only place where your opinion can enter in).
- ***Breadth***

  Choose one programming language (I suggest selecting a research or otherwise experimental language; a good place to start looking is Wikipedia `https://en.wikipedia.org/wiki/Timeline_of_programming_languages`) to explore its impact and evolution throughout its lifetime.
  (1) Briefly introduce it and the historical circumstances of its development (less than a page), including any any specific problem domains it was tailored to work well in, software development issues that it aimed to solve for programmers, or inspirations that lead to its invention.
  (2) Select a number of unique or interesting aspects of your chosen language (such as those listed above as "depth" topics) and discuss how it approaches them. The most important part of discussing the approach is walking through the consequences of the design in the lives of programmers and their software. (Each one should be at least a page and perhaps more.)
  (3) End with a brief personal perspective on what you think its influence has been or should be on the world of programming languages (roughly a page).

Date: Spring 2023.

2. Do's and Don'ts

Here are some tips to write a good research paper for this topic.
DO:

- Use this opportunity to learn about a new programming language, technology, or technique!
  - One goal is to do *research*, which means to find *new* information.
  - It is better to read about something you are interested in but don't already know, and write down what you learned, rather than trying to do it off the top of your head.
- Try to tie in the topic of this paper with your other interests or classes.
  - This class is about the organization of programming languages, so your paper has to cover some aspect of programming language design, technology, theory, history, or application.
  - But because all software is written in some programming language, there are connections to almost every other topic in computer science!
  - Are you interested in graphics or web programming? You could write about how programming languages can support parallel and concurrent processing to harness the power of GPUs and a large network of computers.
  - Are you interested in security? You can write about what language designers have done to help (or hurt) the cause of programmers trying to write secure software systems.
  - Are you interested in embedded programming or cyber-physical systems? You could write about what languages have done to support real-time systems that have to ensure certain steps happen at a specific moment in time (for example, to make sure the motor stops before a robot crashes).

DON'T:

- Write about the only programming language you know.
  - If you only have experience with C/C++, then writing about how C++ is "C with classes" will be an *extremely* hard paper to write well, because you don't have enough familiarity yet with other languages to put them into context in the larger landscape and history of programming languages.
  - If you are going to write about C++, you should be able to compare it to a much more diverse representative collection of languages to explain the design decisions of C++. For example, how does C++ compare to Java? Smalltalk? Modula? Standard ML? Lisp? Forth?
  - Better yet, take this opportunity to write your paper on one of these other languages, and you can compare them to what you know about C/C++!
- Confuse the ecosystem with the language.
  - The Python standard library and the Pip package installer/repository are tools to help you program in Python; they are not the same thing as the Python programming language itself.
  - Writing your paper about how Python is a good programming language because "it has lots of libraries" is like writing a paper explaining

how combustion engines work because "there are lots of gas stations around to fill up on fuel." If the best part of a language is just the amount of its code flying around, then surely Intel x86 assembly is the best because it has the most code!

– If you want to write about how you enjoy Python because it and its libraries make your life easier, focus on what it is about the *language* itself that makes sharing code so easy and enjoyable. C/C++ and Java also have lots of libraries available, but why isn't reusing them as convenient and useful as it is with Python? What kinds of modularity features, type system, and mechanisms for code-reuse does Python provide to help solve this problem better compared to other languages?

3. Policy

Your paper will be due on ***April 28*, the last regular class meeting**. I will not pre-grade your paper through office hours or slack/email, but I can try to discuss your topic with you or specific areas you are struggling. Your will submit your paper inline via Blackboard in the same section as your regular homework assignments by uploading a PDF by the due date listed.

The PDF of your paper must be typed (not handwritten). Typesetting software (like LaTeX) is highly recommended, but you may also use a standard Word processors (like Microsoft word) for preparing your paper as long as you convert your final draft to a PDF for submission.

Your paper must be formatted with 0.5"–1.0" margins, 11pt Times New Roman or Computer Modern font, and double-spaced to 1.5-spaced. Each page must be numbered in the footer, and text must written on white paper with black ink (color may be used in diagrams, or only used sparingly in the text if you get approval). Work that is incorrectly formatted will not be graded and I will consider it as though you never turned it in.

You should not include long quotes, code samples, or large diagrams. I recommend using only mild formatting for emphasis and clarity. Sources should be referenced consistently and accurately through your paper. If you prepare your paper using LaTeX, I recommend organizing your sources automatically using BibTeX with the "alpha" citation style.

You should aim to write about eight (8) pages of double-spaced content, not counting references. Your paper will receive a grade from 0% to 100%. I will evaluate your paper based on the veracity of its content, its logical structure, as well as its grammar, style, and form.